Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes*

Bernd Michelberger*, Bela Mutschler*, and Manfred Reichert†

*University of Applied Sciences Ravensburg-Weingarten, Germany

{bernd.michelberger, bela.mutschler}@hs-weingarten.de

†Institute of Databases and Information Systems, University of Ulm, Germany

manfred.reichert@uni-ulm.de

***Abstract*—Today, enterprises are confronted with a continuously increasing amount of data. Examples of such data include office files, e-mails, process descriptions, and data from process-aware information systems. This data overload makes it difficult for knowledge-workers to identify the information they need to perform their tasks in the best possible way. Particularly challenging is the alignment of process-related information with business processes. In fact, process-related information and business processes are usually managed separately. On the one hand, enterprise content management systems, shared drives, and Intranet portals are used for organizing information, on the other hand, process management technology is used to design and enact business processes. With *process-oriented information logistics* (POIL) this paper presents an approach for bridging this gap. POIL enables the process-oriented and context-aware delivery of process-related information to knowledge-workers. We also present a clinical use case and a proof-of-concept prototype to demonstrate the application and benefits of POIL.**

***Keywords*-process-oriented information logistics, information management, business process management.**

I. Introduction

Market globalization has led to massive cost pressure and increased competition for businesses. Products and services must be developed in ever-shorter cycles and new ways of collaboration within and across enterprises are continuously emerging. As example, consider the treatment of patients in integrated healthcare networks [1].

A major problem is the increasing amount of data enterprises are confronted with [2]. Typical data include, for example, office files, e-mails, web data, informal process descriptions, process models, forms, checklists, best practices, and guidelines. All this enterprise data is provided using shared drives, databases, enterprise applications, Intranet portals, or process-aware information systems. This heterogeneity, both on the data and the data source level, makes data management a time-consuming, complex task.

Moreover, employees do not only need access to data, but require information, i.e., organized data that is used for a specific purpose and in a specific context [3]. In particular, selecting needed information is even more time-consuming and complex than just managing data [4]. Often encountered problems concern incomplete, incorrect or outdated information [5]. Another challenge is the identification of required information to accomplish business processes in the best possible way. To address these challenges, an approach that allows aligning process-related information (we denote such information as *process information* in the following) with business processes is needed, i.e., an approach that enables both *information-* and *process-awareness* [6].

Still, information- and process-awareness are not yet sufficient since the alignment of process information with business processes is strongly influenced by the knowledge-worker's (or process participant's) work context [7]. For example, consider a process description: In a specific work context only selected parts of it might be relevant for a knowledge-worker. Also, less experienced knowledge-workers might need a more detailed process description than experienced ones. Hence, in order to effectively provide the needed information (i.e., the process description), the knowledge-worker's context must also be taken into account, i.e., *context-awareness* must be enabled (cf. Fig. 1) [8].

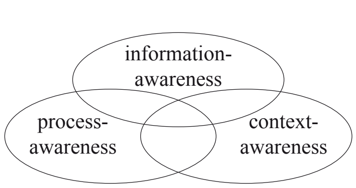

Figure 1. Problem dimensions.

This paper picks up this challenge and suggests *process-oriented information logistics* (POIL), an approach integrating information-, context-, and process-awareness (cf. Fig. 1). Specifically, POIL enables the process-oriented and context-aware delivery of process information to knowledge-workers. It is particularly suitable for knowledge-intensive business processes [9] involving large amounts of process information, user-interaction, and decision-making.

The presented research is performed in the niPRO project.

*This paper was done in the niPRO research project. The project is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 17102X10. More information can be found at http://www.nipro-project.org.

The project's overall goal is to support knowledge-workers with process information depending on their current work context. Key challenges include the delivery of process-oriented, context-aware process information and the flexible, user-oriented visualization of process information.

The remainder of this paper is organized as follows. Section 2 introduces a case from the clinical domain which we use throughout the paper. Section 3 presents our POIL framework. In Section 4, we apply our framework to the case from Section 2. Section 5 summarizes related work. Section 6 concludes with a summary and an outlook.

II. Running Example

This section introduces a case from the clinical domain that will be used throughout the paper. The case (cf. Fig. 3) is based on results of a case study we performed at a large German university hospital [10]. It deals with the prescription, procurement, and administration of drugs. The underlying process of our case is knowledge-intensive, i.e., this process comprises steps such as patient examination and diagnosis, and involves many process information (e.g., patient records, laboratory reports, medical orders, drug stock list etc.), user-interaction (e.g., patient examination, create medical orders, drug administration), and decision-making (e.g., on the drugs to be prescribed for a patient).

The process comprises five roles: *doctor*, *nurse*, *pharmacy*, *accounting*, and *drug supplier*.

During the ward round, the doctor prescribes drugs for a particular patient. Based on that prescription, a nurse updates the patient record accordingly, i.e., the drugs and its apportioning are documented. After the ward round, the prescribed drugs are ordered, i.e., an order form is filled out by the nurse and sent to the hospital pharmacy. The pharmacy checks whether the drugs are available, and - if yes - delivers them to the nurse. If the needed drugs are not available, they are ordered by an assistant from the pharmacy and delivered to the nurse as soon as they become available. If the needed drugs are available, the nurse will prepare them and instruct the patient about their effects. Finally, the doctor examines the effects of the drugs during his next ward round.

Regarding this process, we must distinguish between the *process schema* (as shown in Figure 3) and enacted *process instances* [11]. Likewise, we distinguish between *process information schemas*, *process information instances*, and *abstract process information* (cf. Fig. 2).

Process Information

↙ ↓ ↘

Process Information Schemas | Process Information Instances | Abstract Process Information

Figure 2. Types of process information.

Process information schemas include, for example, any kinds of templates, e.g., for medical reports, order forms, or patient records. Process information instances, in turn, are instantiated process information schemas, e.g., patient-specific medical reports, records, or filled forms. Finally, abstract process information cannot be instantiated. As ex-

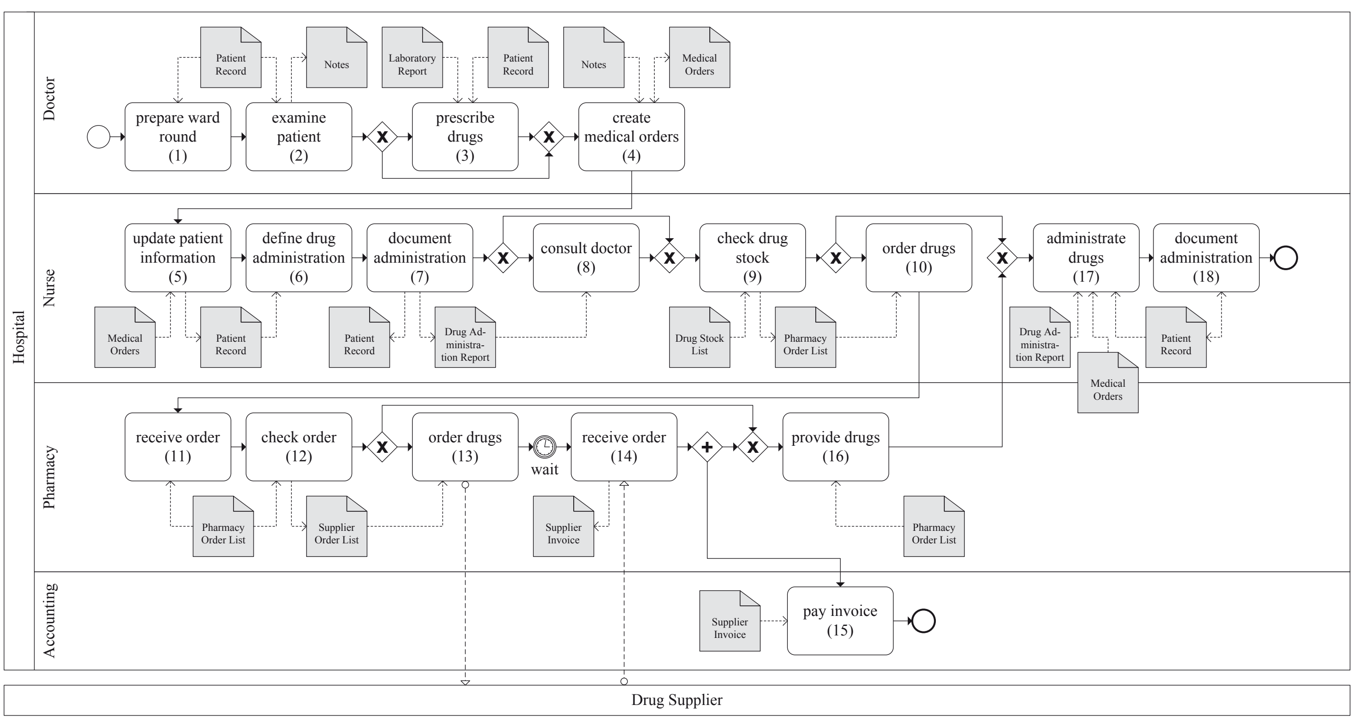

Figure 3. Case: Procurement of drugs.

amples consider working guidelines, lessons learned, and best practices.

Not all types of process information occur in both process schemas and process instances (cf. Tab. I). Process information schemas only occur in process schemas, while process information instances only occur in process instances. Finally, abstract process information can occur in both.

Table I
USE OF PROCESS INFORMATION (PI).

	PI Schema	PI Instance	Abstract PI
Process Schema	×		×
Process Instance		×	×

III. PROCESS-ORIENTED INFORMATION LOGISTICS

Based on the presented process, we now introduce the idea of *information logistics* (IL) in general (Section 3-A), different architectural levels of IL (Section 3-B), and the concepts underlying the notion of a POIL (Section 3-C).

A. Information Logistics

Conventional IL addresses the question how information can be delivered to knowledge-workers as effectively as possible [12]. Information-awareness (e.g., awareness of information quality and information flows) and, to a smaller extent, context-awareness (for the delivery of contextualized information) adopt a key role.

We adopt the notion of IL from [13] and characterize it as follows: IL focuses on the *planning*, *execution*, and *control* of information flows. Its main tasks are the *integration*, *analysis*, and *delivery* of information to individuals taking into account available context information.

IL is independent from the use of information and communication technology (ICT), but ICT, of course, has been intensively used as an IL-enabler in recent years [8]. For example, consider ICT solutions in areas like data warehousing, business intelligence, management information systems, and enterprise content management. However, these solutions suffer from shortcomings, e.g., limited applicability (e.g., only within enterprises) [14], missing operational functionality (as, for example, only the management level is addressed) [15] or from a lack of process orientation.

B. Towards Process-oriented Information Logistics

This section describes five levels of IL. Existing IL approaches (discussed in literature; see below) realize the 1st, 2nd, and 3rd level. Building upon level 3, we introduce two additional levels. The most advanced level (i.e., the 5th level) corresponds to our approach of POIL.

Level 1: Hard-wired Information Logistics. This initial level comprises two architectural layers. A *data layer* manages data sources and an *application layer* delivers process information from the data sources to the users (cf. Fig. 4). Think of an application or a simple Intranet portal that allows users to access process information. Process information and applications are manually linked, i.e., hard-wired and usually based on pre-defined categories such as organizational units, project milestones, or process schemas.

Figure 4 illustrates the manual linking of process information and application. On this level, there is no possibility to realize advanced IL features such as the handling of different levels of process information granularity [5]. Regarding our use case from Section 2, when a doctor prescribes drugs (cf. process step No. 3 in Fig. 3), only entire laboratory reports can be delivered to the doctor, and not parts of it.

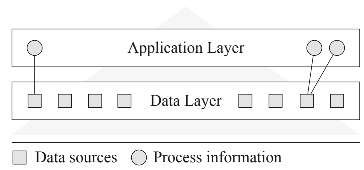

Figure 4. Hard-wired IL (Level 1).

Level 2: Conventional Information Logistics. This second level comprises three architectural layers. Besides the data and application layer, an additional *integration layer* is introduced (cf. Fig. 5). It corresponds to a conventional middleware layer providing a uniform data interface. Still, process information and applications are manually linked and therefore hard-wired.

Based on the integration layer, additional IL features can be realized like the handling of different process information granularity levels [5] or the handling of technical, syntactical, and structural heterogeneity on the data level [16]. In our running example, the provision of certain parts of laboratory reports thus becomes possible. It is still not possible to detect relationships between related process information, e.g., between patient records and related laboratory reports.

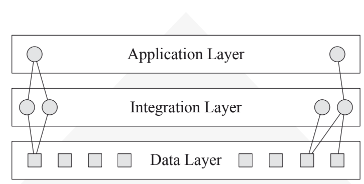

Figure 5. Conventional IL (Level 2).

Level 3: Intelligent Information Logistics. This level also comprises three architectural layers, but the integration layer is replaced by a *semantic integration layer* (cf. Fig. 6). Unlike the conventional integration layer from Level 2, the semantic integration layer does not only realize a uniform data interface, but also provides analysis features for examining integrated process information. In order to apply these analysis features, a *semantic information network* (SIN) is constructed [17]. Similar to an ontology-based model, this

SIN not only comprises *information objects* (i.e., process information), but also relationships between information objects (cf. Section 3-C for details).

Regarding our example, this means that the doctor can be automatically supplied with process information that is related to the process information currently considered. For example, by viewing a laboratory report, the corresponding patient record or related laboratory reports can be automatically determined and delivered to the doctor.

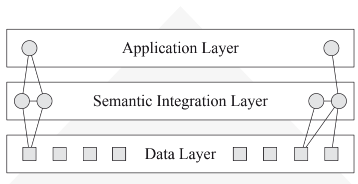

Figure 6. Intelligent IL (Level 3).

Level 4: Context-aware Information Logistics. This level comprises four architectural layers. Besides the data layer, semantic integration layer, and application layer, an additional *context layer* is introduced (cf. Fig. 7).

The purpose of the context layer is to enable the context-aware delivery of process information (cf. Section 1). Therefore, the context layer continuously analyzes a knowledge-worker's situation (or work context) based on available context information [8]. The latter is gathered from different data sources and includes, for example, user name, time, location, and used device. We have described the context layer and its conceptual elements in detail in [7].

Available context information allows filtering the previously discussed SIN (and process information accordingly). This allows, for example, to identify laboratory reports or patient records which are currently needed according to the doctor's work context.

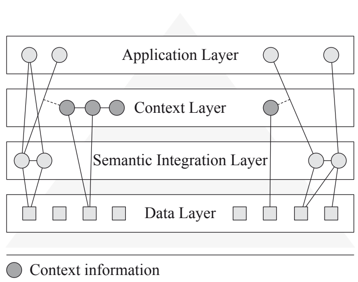

Figure 7. Context-aware IL (Level 4).

Level 5: Process-oriented Information Logistics. This level also comprises four layers. The main difference to Level 4 is the additional consideration of business processes (besides the consideration of process information and context information) (cf. Fig. 8). More precisely, business processes (i.e., process schemas and instances) are integrated into the SIN. This is achieved by splitting them up into their constituent elements (e.g., tasks, gateways, sequence flows, events etc.). Each process element is treated as a single *process object* in the SIN. Hence, the SIN not only contains information objects, but also process objects.

The SIN is enriched and becomes more comprehensive as it includes both process information and business processes. Section 3-C will show that a more effective alignment of process information with business processes (both on the process schema and the process instance level) becomes possible based on this enriched SIN. In our clinical example, a doctor can now be provided with exactly the needed patient records when performing respective process steps (e.g., prescribe drugs) (cf. Section 4 for details).

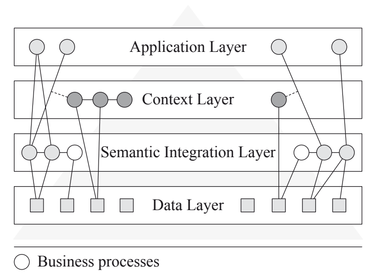

Figure 8. Process-oriented IL (Level 5).

Generally, the goal of POIL is to provide the right process information, in the right format and quality, at the right place, at the right point in time, and to the right people. Therefore, process participants do not have to actively search for relevant process information anymore, but are automatically supplied with needed process information - even if their work context is dynamically changing.

Altogether, POIL combines information-, context-, and process-awareness (cf. Fig. 9). It is *information-aware* as it allows to effectively handle process information and its meaning. It is *context-aware* as it supports the use of context information to characterize the process participant's situation. It is *process-aware* as it allows to integrate and analyze business processes (both process schemas and process instances). The next section describes Level 5 in detail.

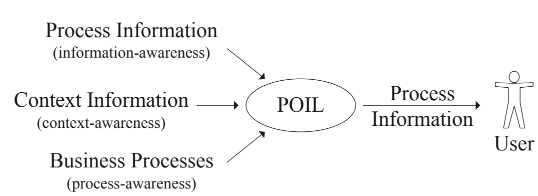

Figure 9. Inputs for POIL.

C. Delivering Contextualized Process Information

As mentioned in Section 3-B, enabling POIL requires four architectural layers: a *data layer*, *semantic integration layer*, *context layer*, and *application layer* (cf. Fig. 10). These layers and their interplay are now described in detail. We put a strong focus on the semantic integration layer as the most important core element of POIL.

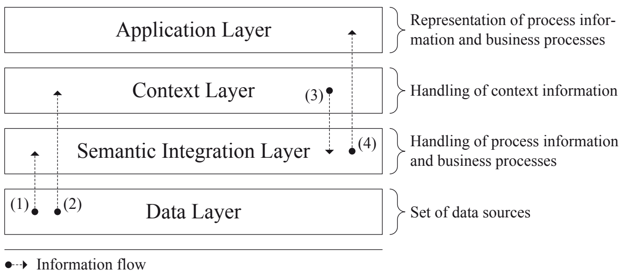

Figure 10. Interplay of POIL architecture levels.

Data Layer. The *data layer* makes the set of data sources to be integrated into POIL available, i.e., the sources of process information, context information, and business processes.

Semantic Integration Layer. The *semantic integration layer* is responsible for the integration and analysis of both process information and business processes. After integrating these into the semantic information network (SIN) (cf. information flow No. 1 in Fig. 10), the SIN is analyzed by using various algorithms (see Phases 3/4). Gathered context information (cf. information flow No. 2) is then used to filter the SIN (cf. information flow No. 3). This enables the semantic integration layer to deliver currently needed process information to the user (cf. information flow No. 4).

The most important core element of POIL is the SIN (constructed by the semantic integration layer). It is constructed following a bottom-up approach and comprises *information objects*, *process objects*, and *relations* between them. Information objects include process information schemas, process information instances, and abstract process information (cf. Section 2). Process objects, in turn, include elements of process schemas and process instances. Each of these objects may be associated with metadata (e.g., acquisition time, file size, and file format). Relations may exist among information objects (e.g., a file which is similar to another one), among process objects (e.g., an event which triggers a task), and between information and process objects (e.g., a file required for the execution of a task). Furthermore, relations are labeled (with the reason of the relationship) and weighted (with the relevance of the relationship) [18]. This allows for determining why objects are related and how strong their relation is.

Figure 11. a) bidirectional and b) unidirectional relations.

Relations can be both bidirectional and unidirectional. Bidirectional edges represent bidirectional relationships between objects, i.e., the relation is valid in both directions (e.g., medical reports have the same author) (cf. Fig. 11a). Unidirectional edges, in turn, represent unidirectional relationships between objects, i.e., the relation is only valid in one direction (e.g., a medical report is a report, but not every report is a medical report) (cf. Fig. 11b).

The SIN is constructed and maintained in six phases:

Phase 1: Integration of Process Objects. In a first step, the process schemas relevant for POIL are integrated. For this purpose, all relevant process objects (e.g., tasks, data objects, sequence flows, gateways etc.) are identified. These objects are then used to create the SIN's first stage of expansion. In a second step, existing process instances of the integrated process schemas are included. Besides the process objects themselves, corresponding metadata such as creation date or tasks deadlines are also considered and associated with the process objects. Some metadata are automatically available (e.g., creation date, content address, modification time) whereas others have to be defined manually (e.g., project milestones, temporal process constraints).

The business processes to be integrated have to be formally specified, e.g., using a formal process modeling language such as the Business Process Modeling Notation (BPMN). Only a formal process representation allows to automatically transform a process schema and corresponding process instances into process objects. Due to space limitations we cannot present the transformation algorithms in detail. However, the steps of such algorithms are similar to the extract, transform, and load steps in data warehousing.

Figure 12 shows the SIN resulting from this first phase for the first three process steps (only process schema) of our use case (cf. Section 2).

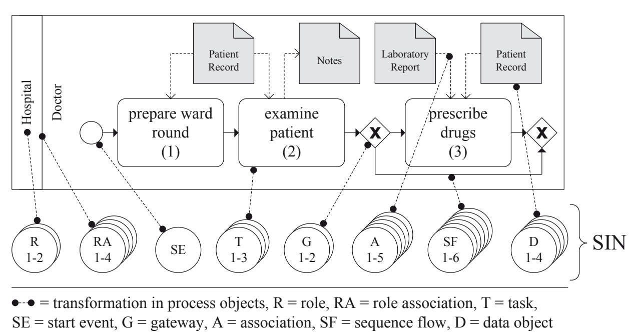

Figure 12. SIN after Phase 1.

Phase 2: Integration of Information Objects. This second phase deals with the integration of process information schemas, process information instances, and abstract process information into the SIN (cf. Fig. 13). Only process information from data sources that are connected by the data layer may be integrated. The already existing SIN (cf. Phase 1) is extended by information objects of different granularity levels, ranging from fine-granular information (e.g., a database tuple, single pages) to coarse-granular information (e.g., database tables, a multi-page document). Like in Phase 1, not only the process information is integrated, but also metadata such as file size or author is attached to the information objects.

Figure 13 shows the resulting SIN for our clinical example with integrated process information (e.g., medical reports, patient records, notes). Note that at this stage, SINs may already include up to hundreds or even thousands of both information objects and process objects.

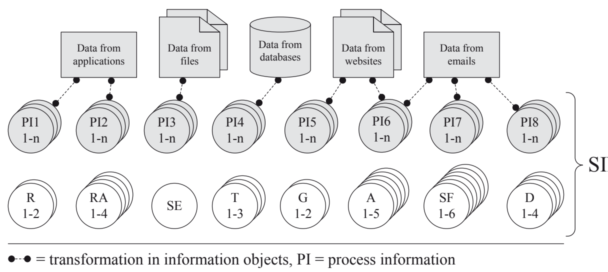

Figure 13. SIN after Phase 2.

Phase 3: Process Object Relationships. This phase deals with the identification of relationships between process objects. Process objects of types *sequence flow*, *association*, *role association*, and *message flow* (using the BPMN terminology) are transformed into relationships, and are then deleted (cf. Fig. 14). Any of these relationships results in an edge between two nodes. Each edge is labeled with a relationship reason, called *relation reason* and a relationship relevance, called *relation weight* (e.g., a double value ranging from 0 to 1 where a higher value indicates higher relevancy). The labeling of relationships allows supporting different scenarios such as "find experts" using the relation reason *author* or the "find related information" using relation reason *text-similarity*. At the end of this phase, all business processes to be included in the POIL are present in the SIN.

Figure 14 shows the SIN at the end of Phase 3. As explained, certain process objects (i.e., SF 1-6, RA 1-4, A 1-5) have been transformed into relationships (i.e., edges).

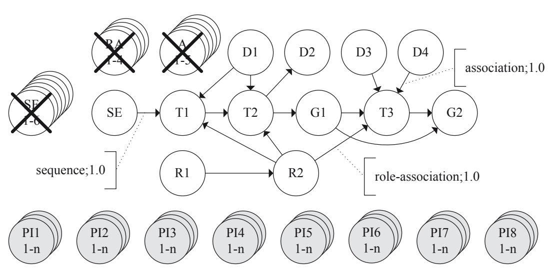

Figure 14. SIN after Phase 3.

Phase 4: Information Object Relationships. This phase deals with the identification of relationships between information objects. Explicit relationships, like hyperlinks or foreign key relationships are identified in the first step. In the second step, algorithms from the fields of data mining, text mining (e.g., text preprocessing, linguistic preprocessing, vector space model, clustering, classification, information extraction) [19], pattern-matching, and machine learning (e.g., supervised learning, unsupervised learning, reinforcement learning, transduction) are applied [20].

More precisely, we use (both syntax- and semantic-based) algorithms such as (inverse) term frequency algorithms, link popularity algorithms, or utilization context algorithms in order to detect the meaning of information objects. In our research, we use a commercial semantic middleware platform implementing these algorithms [18].

As examples of detected relationships between information objects consider metadata matches (e.g., author, keyword, content address), text similarities, utilization context similarities, and cluster similarities. Like in Phase 3, relationships are represented by edges that are labeled with relation reasons and relation weights. Like for process objects, several edges between information objects may exist.

Figure 15 shows the SIN after Phase 4. Note that in actual practice, the number of relationships between information objects can be much higher.

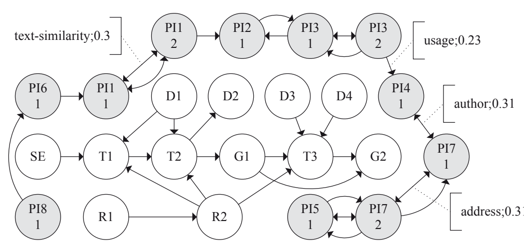

Figure 15. SIN after Phase 4.

Phase 5: Cross-Object Relationships. This phase deals with the analysis of relationships between process objects and information objects. For this purpose the same algorithms as previously used in Phase 4 are reapplied. In addition, metadata matcher and pre-defined business rules are used to detect further relationships, e.g., based on metadata.

Figure 16 shows the SIN at the end of Phase 5. As illustrated, there now exist relationships between process objects (e.g., data objects, tasks, gateways) and information objects (e.g., e-mails, office files, medical reports).

Phase 6: Maintenance. This final phase deals with the continuous integration and analysis of information and process objects. The most important tasks during this phase include the continuous determination of relationships between new and existing objects and validation checks of existing relationships. For this, the semantic integration layer supports both a push and a pull mechanism. With the push mechanism data sources give notifications on changed information and processes. In turn, the pull mechanism is

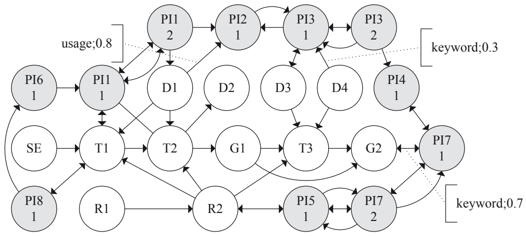

Figure 16. SIN after Phase 5.

based on scheduled integration jobs (e.g., Quartz scheduler). Continuous analysis also includes cleansing of outdated or no more existing information and process objects. Thereby, users may also manually modify the SIN, i.e., they can rate individual objects or create (both public and private) favorites. In summary, Phase 6 deals with the repeated execution of Phases 1 to 5 and additional complex tasks (e.g., the determination of the validity of information and process objects with respect to the maintenance of the SIN).

Figure 17 shows our example from Section 2. Due to space limitations, we limit ourselves to two running process instances and to the process steps that are performed when no drugs have to be ordered. For a better understanding, we also depict some objects (e.g., D1, D4, D5, R2, R3 etc.) twice. These objects can be recognized by a dashed line. Also, for simplicity, role hierarchies are not shown. Note that the SIN is a labeled and weighted Digraph.

Context Layer. The *context layer* is responsible for integrating and analyzing context information. In [7], we have described a framework realizing the context layer. Context information is gathered from data sources called sensors. We distinguish between physical sensors (e.g., thermometer), virtual sensors (e.g., keyboard input), and logical sensors (e.g., sensors which allow to detect a process participant's position by analyzing logins at devices and a mapping to locations) [21]. In addition, further context information can be also derived from existing one (e.g., by aggregation).

A context model, which is constructed based on available context information, allows characterizing a knowledge-worker's work context which can then be used to filter the SIN. The context model is completely independent from the SIN, i.e., context objects are only stored in the context model but not in the SIN. Like the SIN, the context model is a labeled and weighted Digraph (cf Fig. 19).

The context model is constructed in two phases:

Phase 1: Integration of Context Objects. The first phase concerns the integration of all available context information. For this purpose, context objects (e.g., user name, department, e-mail) are identified and gathered from sensors. Context objects are then used to create the context model

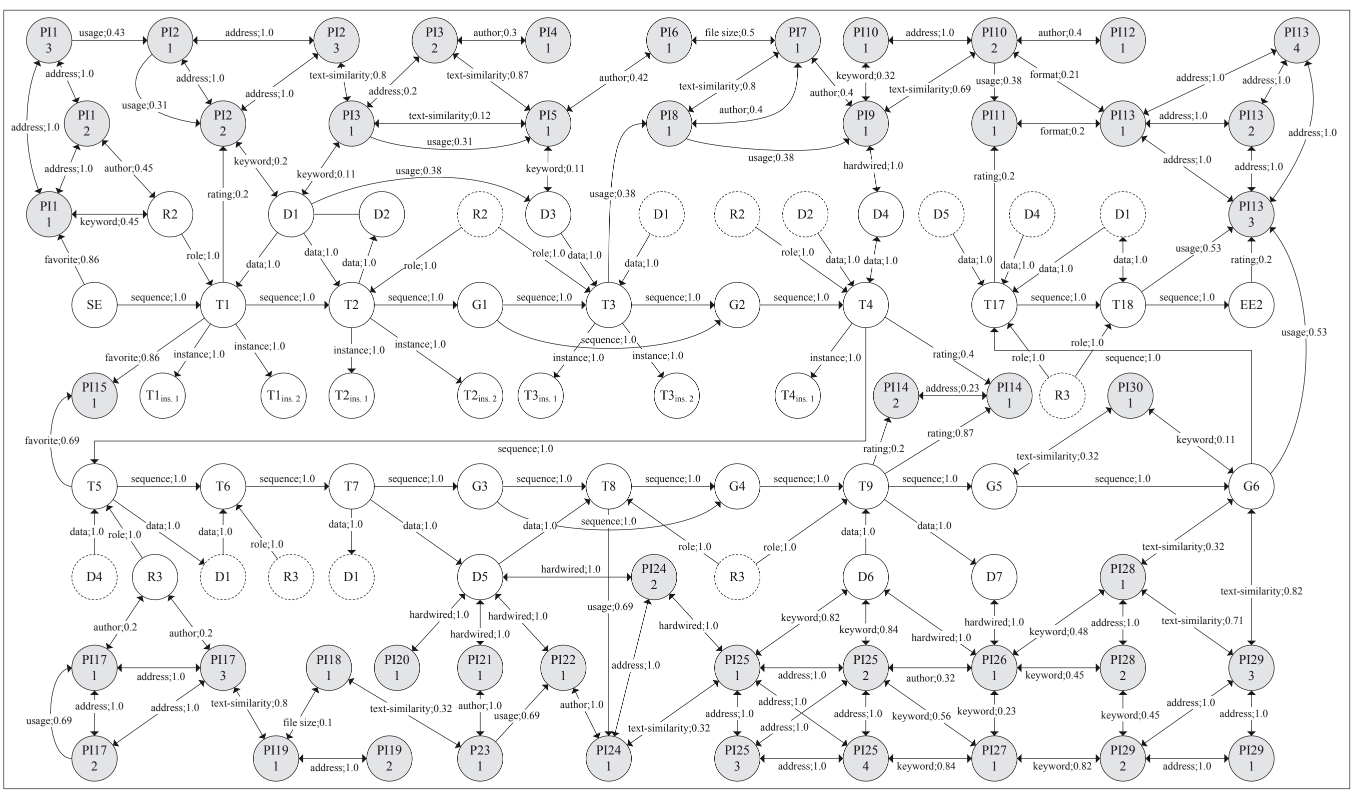

Figure 17. SIN: Procurement of drugs.

(cf. Fig. 18). Note that the validity of context information can rapidly change (e.g., when a user changes his location). Therefore, the context layer must support real-time and rapid processing of context information.

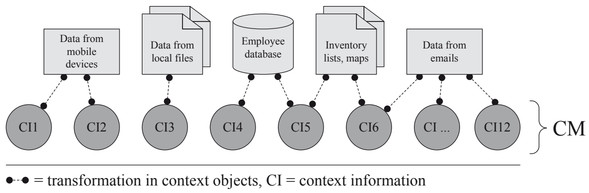

Figure 18. Context model after Phase 1.

Phase 2: Context Objects Relationships. The second phase concerns the analysis of context objects, i.e., the identification of relationships between them. For this purpose, algorithms enabling the aggregation, interpolation, and interpretation of context objects are used [22]. For example, instead of Global Positioning System (GPS) coordinates, the room number is calculated (aggregation) or incomplete context information is completed (interpolation).

Figure 19 shows an examplary context model (CM) based on context factors (e.g., user, location, time, environment) as introduced in [7] for the following situation: "Doctor Peter Miller examines a patient on Monday, 27th February, 2012, in room number 301 using a tablet computer".

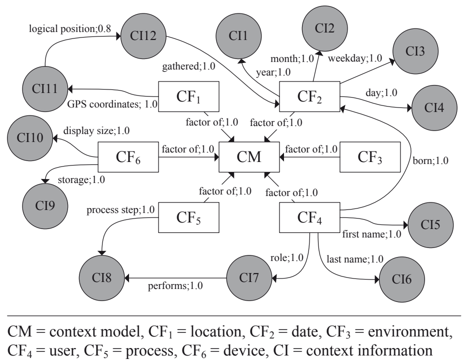

Figure 19. Context model after Phase 2.

Application Layer. Finally, the *application layer* is responsible for the joint presentation and delivery of process objects, information objects, and context objects to users.

IV. Applying our Approach

Along three use cases, we demonstrate how POIL can support knowledge-workers with personalized and contextualized process information.

Use Case 1. Consider the second process step of our clinical use case (i.e, the patient examination) for which the doctor needs access to patient records. In actual practice, patient records include extensive and various kinds of data (e.g., master and transaction data, department-specific data, historical data). However, only small parts of a record (e.g., former diseases, pre-existing conditions, course of disease) are actually needed in the context of a patient examination. POIL is able to provide the needed parts.

The underlying SIN is constructed in the above mentioned six phases: First, the formalized business process, i.e., our clinical business process (cf. Section 2), is integrated (cf. Phase 1). Second, needed process information, i.e., the patient records, are also integrated (cf. Phase 2). After that, relationships among process objects (cf. Phase 3) or information objects (cf. Phase 4) and between process objects and information objects (cf. Phase 5) are determined. Finally, the SIN is maintained (cf. Phase 6). Performing these steps is the prerequisite to handle this use case.

In order to provide personalized and contextualized process information the context model is then constructed in the two mentioned phases (cf. Section 3-C). The construction of the context model is initiated at specific points in time (e.g., by a scheduled job) or when a user performs a certain task (e.g., patient examination).

Let us assume that the doctor is on his ward round: Based on the context layer, the doctor's work context can be determined. For this purpose, location identification technologies (e.g., satellite networks, cellular networks or indoor networks) can be used. Having identified the doctor's location, the technical position (e.g., GPS coordinates) is mapped to a logical position (e.g., a room number) using a hospital building map. Analogous to the integration and analysis of the doctor's location, other context information such as time, device or role may be considered. This additional context information (e.g., bed occupancy) allows determining which patients are staying in which room.

Combining this context information with process information, the doctor can be provided with relevant patient records according to his location. For example, if patient Henry Jonson is in Room 301 and the doctor enters the room, he automatically gets the patient record of Henry Jonson (e.g., on his tablet). The granularity level of the patient record depends on the user's role and the current process step, i.e., only parts of the patient record are provided which are necessary for the doctor when examining a patient.

Use Case 2. Our second use case is based on the third process step of our clinical process, i.e., the prescription of drugs. This process step is highly knowledge-intensive (due to questions like which disease has the patient and which treatment should he get) and includes, among others tasks, the interpretation of symptoms (e.g., excessive thirst, tiredness), the determination of diseases (e.g., diabetes mellitus type 2), and the identification of treatment options (e.g., physical activity, balanced diet, prescription of drugs).

POIL can support these tasks: For this purpose, we assume that different process information about symptoms, diseases, treatment options, and drugs are integrated in the SIN. Data sources are, for example, digital medical libraries or health web portals (e.g., WebMD). Without POIL, relationships between symptoms, diseases, and treatment options are not given. With POIL, the semantic integration layer determines relationships and meaning of process information (e.g., which symptom belongs to which disease and how to treat the disease) (cf. Phase 4). After the doctor has entered symptoms, POIL is able to make suggestions which diseases the patient may have and which treatment options exists.

For this second use case we have developed a proof-of-concept web application based on Java and a semantic middleware [18] (cf. Fig. 20). A screencast explaining the web application can be found on our project website[1].

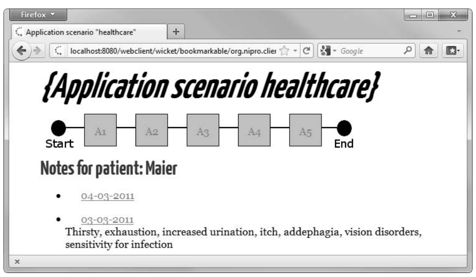

Figure 20. Screenshots niPRO healthcare prototype.

Use Case 3. This third example is based on process step No. 17, i.e., the administration of drugs. To perform this process step, certain process information is needed: a drug administration report, a medical order, and a patient record. The range of available data sources and process information often makes it difficult for nurses to find needed information. When they look for information, they often do not know which information is needed and where the information can be found. Therefore, they often need a very long time to get the needed information.

POIL may accomplish this task. By integrating business processes (both on the process schema and process instance level, cf. Phase 1 in Section 3-C), process information can be effectively aligned (cf. Phases 5/6) with business processes. We assume that for all considered process instances in the POIL the corresponding process schemas are available.

If the context layer determines that a nurse currently administrates drugs (by the use of the context model), POIL can automatically provide the needed process information to the nurse (e.g., in a Intranet portal).

Besides the mentioned use cases, POIL supports many other use cases such as process monitoring, time management, expert finding, collaboration, and decision support.

V. Related Work

Various authors investigate information logistics (IL) in general. Bucher and Dinter [13] conduct, for example, an empirical study and assess benefits, design factors, and realization approaches for IL. Heuwinkel and Deiters [12] demonstrate the possibilities and advantages of IL in the healthcare sector. Technology enablers and management challenges related to IL are discussed by Winter [15]. Context-awareness in IL is discussed, for example, by Haseloff [8] and Meissen et. al [23].

Haftor et. al [24] have conducted a comprehensive literature study in the field of information logistics and identify eleven research directions. Two of these research directions are of particular relevance for us: the user-driven information supply [12] and the supply of analytical information [14].

As mentioned in Section 1, various ICT solutions have been proposed to enable IL, including, for example, data warehousing (DWH), business intelligence (BI), decision support systems (DSS), and enterprise content management (ECM). However, these approaches suffer from weaknesses (cf. Tab II). DWH, for example, rather focusses on the creation of an integrated database [25]. Traditional BI, in turn, addresses data analytics and is typically completely isolated from business processes execution [13]. Conventional DSS support complex business decision-making, i.e., they serve the management level [26]. In contrast, ECM deals with the management of information across enterprises referring to related strategies, methods, and tools [27].

Table II summarizes which parts of a POIL are supported by the above mentioned approaches.

Table II
Approaches to enable POIL.

	information-awareness		context-awareness	process-orientation		delivery of PI	
	int.	ana.		ope.	str.	kno.	dec.
DWH	×	(×)	–	(–)	(–)	(×)	(×)
BI	(×)	×	–	(–)	(–)	(×)	(×)
DSS	(×)	×	–	–	(–)	–	(×)
ECM	×	×	–	(×)	(×)	(×)	(×)

× = main focus, (×) = is supported, (–) = not generally supported, – = not supported, int. = integration, ana. = analysis, ope. = operational, str. = strategical, kno. = knowlege-worker, dec. = decision-maker

VI. Summary and Outlook

Enterprises are confronted with an ever increasing amount of data. A major problem in practice is the alignment of process-related information with business processes. So far, they are usually handled separately, e.g., through shared drives and information portals on the one hand and through process management technology on the other hand.

This paper suggests a new approach to bridge this gap, called *process-oriented information logistics*. Specifically, the contribution of this paper is threefold: First, we sketch the path from conventional information logistics to innovative, process-oriented information logistics. Second, we

[1] http://nipro.hs-weingarten.de/screencast

introduce basic concepts of process-oriented information logistics. Third, we demonstrate the application and benefits of our approach along three clinical use cases.

Future research will include a more detailed investigation of handling process schemas and process instances within semantic information networks as well as of performance and scalability issues. We are also developing further proof-of-concept applications realizing the concepts presented in this paper.

References

[1] R. Lenz and M. Reichert, *IT Support for Healthcare Processes - Premises, Challenges, Perspectives*. in: Data and Knowledge Engineering, 61(1), pp. 39-58, 2007.

[2] A. Edmunds and A. Morris, *The Problem of Information Overload in Business Organisations: A Review of the Literature*. in: Int'l J. of Information Management, 20(1), pp. 17-28, 2000.

[3] P. Bocij, D. Chaffey, A. Greasley, and S. Hickie, *Business Information Systems: Technology, Development and Management for the E-Business*. Prentice Hall, 2006.

[4] J. Rowley, *The wisdom hierarchy: representations of the DIKW hierarchy*. in: J. of Information Science, 33(2), pp. 163-180, 2006.

[5] B. Michelberger, B. Mutschler, and M. Reichert, *Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter*. in: Proc. 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107-120, 2011.

[6] V. Künzle and M. Reichert, *PHILharmonicFlows: Towards a Framework for Object-aware Process Management*. in: J. of Software Maintenance and Evolution: Research and Practice, 23(4), pp. 205-244, 2011.

[7] B. Michelberger, B. Mutschler, and M. Reichert, *A Context Framework for Process-oriented Information Logistics*. in: Proc. 15th Int'l Conf. on Business Information Systems (BIS'12), LNBIP 117, pp. 260-271, Vilnius, 2012.

[8] S. Haseloff, *Context Awareness in Information Logistics*. PhD Thesis, Technical University of Berlin, 2005.

[9] N. Gronau, C. Müller, and R. Korf, *KMDL - Capturing, Analysing and Improving Knowledge-Intensive Business Processes*. in: J. of Universal Computer Science (JUCS), 11(4), pp. 452-472, 2005.

[10] B. Michelberger, B. Mutschler, and M. Reichert, *On Handling Process Information: Results from Case Studies and a Survey*. in: Proc. 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333-344, Clermont-Ferrand, 2011.

[11] S. Rinderle, M. Reichert, and P. Dadam, *On Dealing with Structural Conflicts between Process Type and Instance Changes*. in: Proc. 2nd. Int'l Conf. Business Process Management (BPM'04), pp. 274-289, Potsdam, 2004.

[12] K. Heuwinkel and W. Deiters, *Information logistics, e-healthcare and trust*. in: Proc. Int'l Conf. e-Society (IADIS'03), 2, pp. 791-794, Lisbon, 2003.

[13] T. Bucher and B. Dinter, *Process Orientation of Information Logistics - An Empirical Analysis to Assess Benefits, Design Factors, and Realization Approaches*. in: Proc. 41st Annual Hawaii Int'l Conf. on System Sciences, pp. 392-402, 2008.

[14] B. Dinter and R. Winter, *Information Logistics Strategy - Analysis of Current Practices and Proposal of a Framework*. in: Proc. 42nd Hawaii Int'l Conf. on System Sciences (HICSS-42), pp. 1-10, Hawaii, 2009.

[15] R. Winter, *Enterprise-wide Information Logistics: Conceptual Foundations, Technology Enablers, and Management Challenges*. in: Proc. 30th Int'l Conf. on Information Technology Interfaces (ITI'08), pp. 41-50, Dubrovnik, 2008.

[16] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*. Addison-Wesley, 2010.

[17] J. E. Sowa, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, 1991.

[18] J. Wurzer and B. Mutschler, *Bringing innovative Semantic Technology to Practice: The iQser Approach and its Use Cases*. in: Proc. 4th Int'l Workshop on Applications of Semantic Technologies (AST'09), pp. 3026-3040, 2009.

[19] A. Hotho, A. Nürnberger, and G. Paa, *A Brief Survey of Text Mining*. in: J. for Computational Linguistics and Language Technology, 20(1), pp. 19-62, 2005.

[20] J. Wurzer, *New Approach for Semantic Web by Automatic Semantics*. in: European Semantic Technology Conf. (ESCT'08), Vienna, 2008.

[21] J. Indulska and P. Sutton, *Location Management in Pervasive Systems*. in: Proc. Information Security Workshop Conf. on ACSW Frontiers'03, 21, pp. 143-151, Adelaide, 2003.

[22] A. K. Dey, *Providing Architectural Support for Building Context-Aware Applications*. PhD Thesis, Georgia Institute of Technology, 2000.

[23] U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried, *Context- and Situation-Awareness in Information Logistics*. in: Current Trends in Database Technology - EDBT'04 Workshops, pp. 335-344, 2005.

[24] D. M. Haftor, M. Kajtazi, and A. Mirijamdotter, *A Review of Information Logistics Research Publications*. in: Proc. 3rd Workshop on Information Logistics and Knowledge Supply (ILOG 2010), pp. 244-255, 2010.

[25] J. Lechtenbörger, *Data warehouse schema design*. Infix Akademische Verlagsgesellschaft Aka GmbH, PhD Thesis, University of Münster, 2001.

[26] V. S. Janakiraman and K. Sarukesi, *Decision Support Systems*. Prentice-Hall, 2004.

[27] S. A. Cameron, *Enterprise Content Management: A Business and Technical Guide*. British Informatics Society, 2011.